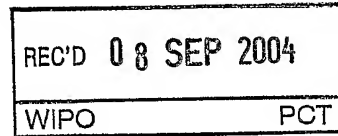


证 明

本证明之附件是向本局提交的下列专利申请副本



申 请 日： 2003.12.31

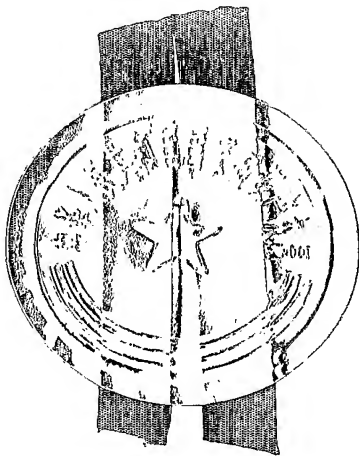
申 请 号： 2003101242014

申 请 类 别： 发明

发明创造名称： 一种在数据库里快速定位数据页中记录的方法

申 请 人： 中兴通讯股份有限公司

发明人或设计人： 李世亮、高洪、洪玲



**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

中华人民共和国
国家知识产权局局长

王 景 川

2004 年 7 月 9 日

1. 一种在数据库里快速定位数据页中记录的方法，其特征在于，包括如下步骤：

5 (1) 在数据页的末端设置一个目录结构，该目录结构由一组记录偏移构成，记录偏移是某条记录在页里的位置偏移；该目录结构中的每个目录称之为 dir_slot，每个 dir_slot 存放一个记录位置的偏移；

(2) 采用定位算法在 dir_slot 中查寻相关记录，在定位到某个 dir_slot 后，根据该 dir_slot 中存放的记录偏移，顺序查找这相关的这一组记录，准确
10 地定位到要找的那条记录。

2. 如权利要求 1 所述的在数据库里快速定位数据页中记录的方法，其特征在于，还包括如下步骤：待查记录放在字段结构体里，数据页中的记录将与字段结构体进行比较。

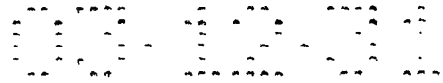
3. 如权利要求 2 所述的在数据库里快速定位数据页中记录的方法，其特征
15 在于，首先把代表 dir_slot 序号的两个变量 low, up 赋初值，low 赋值为 0，up 初值为该数据页上的 dir_slot 的总数，然后进行定位算法查询，判断该记录属于哪个 dir_slot。

4. 如权利要求 1、2 或 3 所述的在数据库里快速定位数据页中记录的方法，其特征在于，所述定位算法为二分法。

20 5. 如权利要求 4 所述的在数据库里快速定位数据页中记录的方法，其特征在于，所述二分法是连续取中间值与字段结构体比较，直到 up-low 的值小于等于 1 为止。

6. 如权利要求 3 或 5 所述的在数据库里快速定位数据页中记录的方法，其特征
25 在于，找到记录后，从序号为 low 的 dir_slot 中，顺序取记录和字段结构体进行比较，直到该记录的下一条记录为序号为 up 的 dir_slot 上的首记录 up_rec；如果在这个过程中找到记录，则在该页完成查找；如果找不到，则转到下一页进行同样的匹配。

7. 如权利要求 1 所述的在数据库里快速定位数据页中记录的方法，其特征
30 在于，在数据库中的一个数据页面上插入一条记录而导致 dir_slot 上记录数满时，则把当前的 dir_slot 分裂成两个，以增加 dir_slot。



8. 如权利要求 7 所述的在数据库里快速定位数据页中记录的方法，其特征在于，该记录插入链表后，该记录所在的 dir_slot 上的记录总数超过最大限值，则把该 dir_slot 后的 dir_slot 全部后移一位，这样就增加了一个 dir_slot，并把该记录所在的 dir_slot 上的所有记录一分为二，把两部分记录分别归属到这两个 dir_slot 上。

9. 如权利要求 1 所述的在数据库里快速定位数据页中记录的方法，其特征在于，在删除一条记录时，把该记录从链表上取下来，并置删除标志。

10. 如权利要求 9 所述的在数据库里快速定位数据页中记录的方法，其特征在于，首先获取该 dir_slot 后面的一个 dir_slot，并判断后面的 dir_slot 上的记录数，如果记录数大于最小限值，就从后一个 dir_slot 上取一条记录，加入当前的 dir_slot 中去；如果记录数小于等于最小限值，就把两个 dir_slot 合并，并删除当前的 dir_slot。

一种在数据库里快速定位数据页中记录的方法

5 技术领域

本发明涉及一种数据库技术中对数据页里记录进行管理的方法，特别涉及一种在数据库里快速定位数据页中记录的方法。

背景技术

- 10 数据库系统是对大量数据进行管理的一种非常有效的软件系统。数据库中最小的管理单位是记录，每条记录记载一组相关信息。数据页是存放记录的物理单元，在一个页面可以存放多条记录。数据页中的每个记录都有一个指针，该指针指向下一条记录，整个页面里的记录都链成一个线性记录链，当搜索记录时，顺着该条线性记录链，就可以定位到具体的某条记录。这种方法的缺陷
- 15 在于：页面搜索效率很低，导致数据库查询效率较低。

发明内容

本发明的目的在于提出一种在数据库里快速定位数据页中记录的方法，能够提高在数据库里定位数据记录的速度。

- 20 数据页中的记录都是顺序存放的，本发明采用的方案如下：在数据页的末端设置一个目录结构，该目录结构由一组记录偏移构成，记录偏移是某条记录在页里的位置偏移。该目录结构中的每个目录称之为 dir_slot，每个 dir_slot 存放一个记录位置的偏移，根据该位置偏移，可以立即定位到一条记录。但并不是每条记录的位置偏移都记录在 dir_slot 中，在数据页中的这个线性记录
- 25 链中，每隔一定的记录数（该记录数在 dir_slot 的记录最大限值和最小限值之间），取一条记录的偏移存放在 dir_slot 中。这样，每个页面就有了一个目录结构，在进行查询时，并不对具体记录进行查找，而是采用快速的定位算法在 dir_slot 中查找相关记录，在定位到某个 dir_slot 后，根据该 dir_slot 中存放的记录偏移，顺序查找这相关的这一组记录，通过这种方法就能够准确
- 30 地定位到要找的那条记录。

具体地讲，本发明公开了一种在数据库里快速定位数据页中记录的方法，包括如下步骤：

(1)在数据页的末端设置一个目录结构，该目录结构由一组记录偏移构成，记录偏移是某条记录在页里的位置偏移；该目录结构中的每个目录称之为
5 dir_slot，每个 dir_slot 存放一个记录位置的偏移；

(2)采用定位算法在 dir_slot 中查找相关记录，在定位到某个 dir_slot 后，根据该 dir_slot 中存放的记录偏移，顺序查找这相关的这一组记录，准确地定位到要找的那条记录。

所述在数据库里快速定位数据页中记录的方法，还包括待查记录放在字段
10 结构体里，数据页中的记录将与字段结构体进行比较。

所述在数据库里快速定位数据页中记录的方法，首先把代表 dir_slot 序号的两个变量 low，up 赋初值，low 赋值为 0，up 赋值为页上的 dir_slot 的总数，然后进行定位算法查询，判断该记录属于哪个 dir_slot。

所述定位算法为二分法。

15 所述二分法查询是连续取中间值与字段结构体比较，直到 up-low 的值不大于 1 为止。

找到记录后，从序号为 low 的 dir_slot 中，顺序取记录和字段结构体进行比较，直到该记录的下一条记录为序号为 up 的 dir_slot 上的首记录 up_rec；如果在这个过程中找到记录，则在该页完成查找；如果找不到，则转
20 到下一页进行同样的匹配。

所述的在数据库里快速定位数据页中记录的方法，在数据库中的一个数据页面上插入一条记录而导致 dir_slot 上记录数满时，则把当前的 dir_slot 分裂成两个，以增加 dir_slot。

该记录插入链表后，该记录所在的 dir_slot 上的记录总数超过最大限值，
25 则把该 dir_slot 后的 dir_slot 全部后移一位，这样就增加了一个 dir_slot，并把该记录所在的 dir_slot 上的所有记录一分为二，把两部分记录分别归属到这两个 dir_slot 上。

所述的在数据库里快速定位数据页中记录的方法，在删除一条记录时，把该记录从链表上取下来，并置删除标志。

30 首先获取该 dir_slot 后面的一个 dir_slot，并判断后面的 dir_slot 上

的记录数，如果记录数大于最小限值，就从后一个 dir_slot 上取一条记录，加入当前的 dir_slot 中去；如果记录数小于等于最小限值，就把两个 dir_slot 合并，并删除当前的 dir_slot。

本发明有益效果在于，相比于现有技术，本发明在一个页面中定位一条记录的速度得到了极大的提高。查询某一条记录时，不需要按记录链顺序查找比较，而是在目录机构中对 dir_slot 进行快速定位查找，这样，节省了大量的顺序查找的开销，在定位到具体的 dir_slot 后，最大的查询比较次数就是 dir_slot 的最大限值记录数。采用该方法大大地节省了查询比较的次数。由于把 dir_slot 放在页的末端，不用在页中预留空间，同时也非常有效地管理了页面记录。由于 dir_slot 只是存放一条记录的偏移，因此占用的空间极少。

附图说明

- 图 1 是本发明的数据页的结构说明；
- 图 2 是本发明的 dir_slot 的增加图；
- 图 3 是本发明的 dir_slot 的删除图；
- 图 4 是本发明在数据页中定位记录的流程图。

具体实施方式

- 图 1 为数据页的整体结构图，它描述了一个数据页的完整的结构。在该图中，前 26 个字节描述该页中记录的属性，26—36 字节描述该页的属性，36—56 是段指针，dir_slot 从页的末尾向上扩展，采用这种方案巧妙地避免了我们为 dir_slot 预留空间。这样，在增减记录时，不必考虑当前存储了多少记录，使用了多少 dir_slot。
- 图 2 是 dir_slot 的增加图，描述了在数据库中的一个数据页面上插入记录时，如果该记录所在的 dir_slot 上记录数已经达到最大限值后，dir_slot 如何把当前的 dir_slot 分裂成两个，从而达到增加 dir_slot 的目的。每页中的记录是一个记录链表，在插入记录时，把该记录插入到该链表的相关位置，一般按升序排列。如图所示，插入链表后（步骤 201），首先获取该记录所在的 dir_slot 上的记录数（slot 号为 slot_no）（步骤 202），然后判断该记录



所在的 `dir_slot` 上的记录数是否超最大限值 (步骤 203), 如果不超过最大限值, 直接记录插入日志并结束 (步骤 212); 如果超过最大限值, 获取该 `dir_slot` 在该页上的地址 `slot` (步骤 204), 获取该 `dir_slot` 上的记录数 `n_owned` (步骤 205), 获取前一个 `dir_slot` 的地址 `prev_slot` (步骤 206), 根据 `prev_slot` 值得到该 `prev_slot` 上的记录指针 (步骤 207), 取得该记录下面的第 `prev_slot/2` 个记录的指针 `recptr4` (步骤 208), 把大于等于 `slot_no` 的 `dir_slot` 后移一位 (步骤 209), 这样就增加了一个 `dir_slot`, 并把该记录所在的 `dir_slot` 上的所有记录一分为二, 即设置 `slot_no` 上的 `dir_slot` 的记录数为 `n_owned/2`, 并把 `dir_slot` 上的记录偏移记为 `recptr4` (步骤 210), 设置 `slot_no+1` 上的 `dir_slot` 的记录数为 `n_owned-n_owned/2` (步骤 211), 这样就可以把两部分记录分别归属到这两个 `dir_slot` 上, 然后记录插入日志并结束 (步骤 212)。

图 3 是 `dir_slot` 的删除图, 描述了当删除记录时, 在 `dir_slot` 上记录数小于最小限值时, 如何合并两个 `dir_slot`。在数据库中的一个数据页面上删除一条记录时, 系统是如何对 `dir_slot` 进行调整的。每页中的记录是一个记录链表, 在删除一条记录时, 把该记录从链表上取下来, 并置删除标志 (步骤 301)。然后取该记录所在的 `dir_slot` 上的记录总数 (步骤 302), 如果记录总数小于等于最小限值 (步骤 303), 那么就进行对 `dir_slot` 的调整工作。首先获取该 `dir_slot` 后面的一个 `dir_slot` (步骤 304—306), 并判断后面的 `dir_slot` 上的记录数 (步骤 307), 如果记录数大于最小限值, 就从后一个 `dir_slot` 上取一条记录, 加入当前的 `dir_slot` 中去, 具体地讲, 取当前 `dir_slot` 的记录指针 `old_rec` (步骤 310), 取该记录的下一条记录指针为 `new_rec` (步骤 311), 设置当前 `dir_slot` 的记录指针为 `new_rec` (步骤 312) 设置当前 `dir_slot` 和后一个 `dir_slot` 的记录为新值 (步骤 313), 然后记录删除日志并结束 (步骤 314)。如果记录数小于等于最小限值, 把该 `dir_slot` 后的所有 `dir_slot` 前移一位 (步骤 308), 合并该 `dir_slot` 和后面的 `dir_slot` (步骤 309), 记录删除日志并结束 (步骤 314)。这样就对 `dir_slot` 进行了调整。

图 4 是在数据页中定位记录的流程图, 它描述了如何在一个页面中定位一条记录, 在数据页上查询一条记录的流程图。待查记录的部分域的值放在字段

结构体 `turple` 里 (步骤 401), (所谓字段结构体 `turple` 是待查记录的部分字段组成的一个结构体。要在数据库中查询一条记录, 必须知道该记录的部分内容, 例如一个人事档案数据库, 通过姓名字段可以进行查询, 姓名字段就构成了一个 `turple`), 数据页中的记录将与字段结构体进行比较。首先把代表

5 `dir_slot` 序号的两个变量 `low`, `up` 赋初值, `low` 赋值为 0, `up` 赋值为页上的 `dir_slot` 的总数 (步骤 402), 然后进行二分法查询, 判断该记录属于哪个 `dir_slot`。二分法查询的方法是不断地取中间值与字段结构体比较, 直到 `up-low` 的值不大于 1 为止。具体地讲, 所述二分法是取页上的 `dir_slot` 的中间值的记录与 `turple` 比较。首先设置 `mid=(low+up)/2`, 据此获取序号为 `mid`

10 的 `dir_slot` 上的记录 `mid_rec`, 将 `mid_rec` 与字段结构体进行比较, 如果 `mid_rec` 大于 `turple`, 则令 `up=mid`, 如果 `mid_rec` 小于 `turple`, 则令 `low=mid`, 重新比较 (步骤 403、404、405、406、407、409)。找到记录后, 从序号为 `low` 的 `dir_slot` 中顺序取记录和字段结构体进行比较, 直到该记录的下一条记录为 `up_rec` (`up_rec` 是序号为 `up` 的 `dir_slot` 上的首记录) (步骤 410、411、

15 412、413、414、415、417)。如果在这个过程中找到记录, 则在该页完成查找 (步骤 408 和步骤 416)。如果找不到, 则转到下一页进行同样的匹配 (步骤 418)。从这个过程中, 通过 `dir_slot` 这种结构, 在页面上能够非常迅速地查找到相关的记录。

例如, 假设一个页面存放了 300 条记录, 如果顺序查找, 需要进行 300

20 次匹配。而如果采用本发明中所描述的方法, 大约需要 40 个左右的 `dir_slot` 存放部分记录的偏移, 采用二分法定位, 最多需要 5 次匹配定位出具体的 `dir_slot`, 在 `dir_slot` 中最多需要定位 8 次, 最坏情况下总共需要定位 13 次, 在页面的查询速度提高了 23 倍。由于把 `dir_slot` 放在页的末端, 不用在页中预留空间, 同时也非常有效地管理了页面记录。由于 `dir_slot` 只是存放

25 一条记录的偏移, 因此占用的空间极少, 按每个偏移 4 个字节算起, 300 条记录共需约 160 个字节左右的存储空间。

说明书附图

SZ01-03P100316

0				8
2 字节 dir_slot 的总数	2 字节 heap_top 的指针	2 字节 在 heap 中的记录数	2 字节 空记录链的首地址	
2 字节 删除记录的字节数	2 字节 最后插入记录的指针	2 字节 最后插入记录的 slot	2 字节 连续插入的记录数	
2 字节 该页上的记录总数	8 字节 最大的事务 id 值			
最大的事务 id 值	2 字节 索引树的高度	8 字节 该页所属的索引树的 id 值		
...该页所属的索引树的 id 值		10 字节 叶子节点所属的段的指针		
... 叶子节点所属的段的指针				
10 字节 ... 非叶子节点所属的段的指针				
.....页里的数据.....				
.....dir_slot 目录字段.....				
8 字节的页校验数据				

图 1

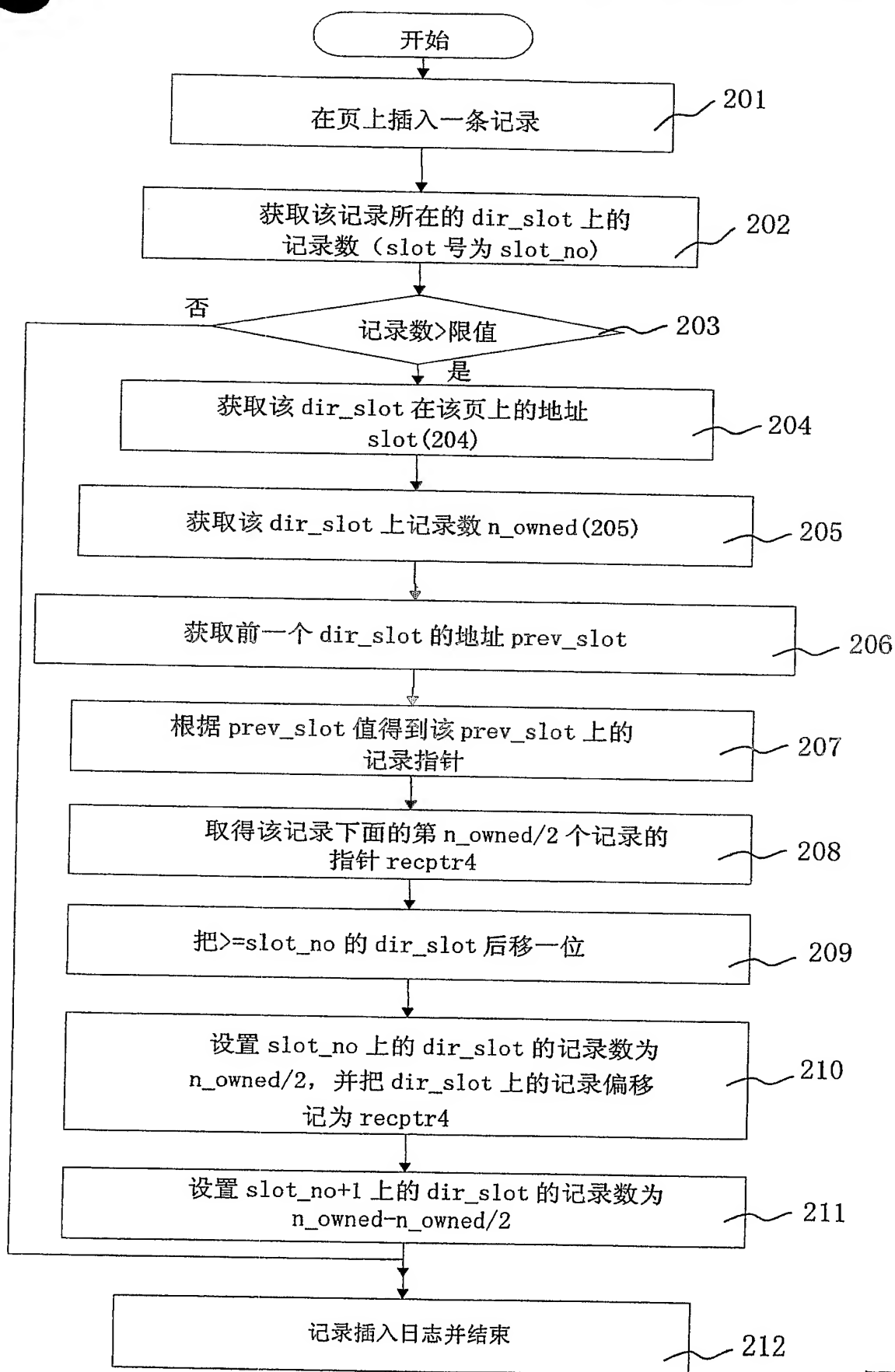


图 2

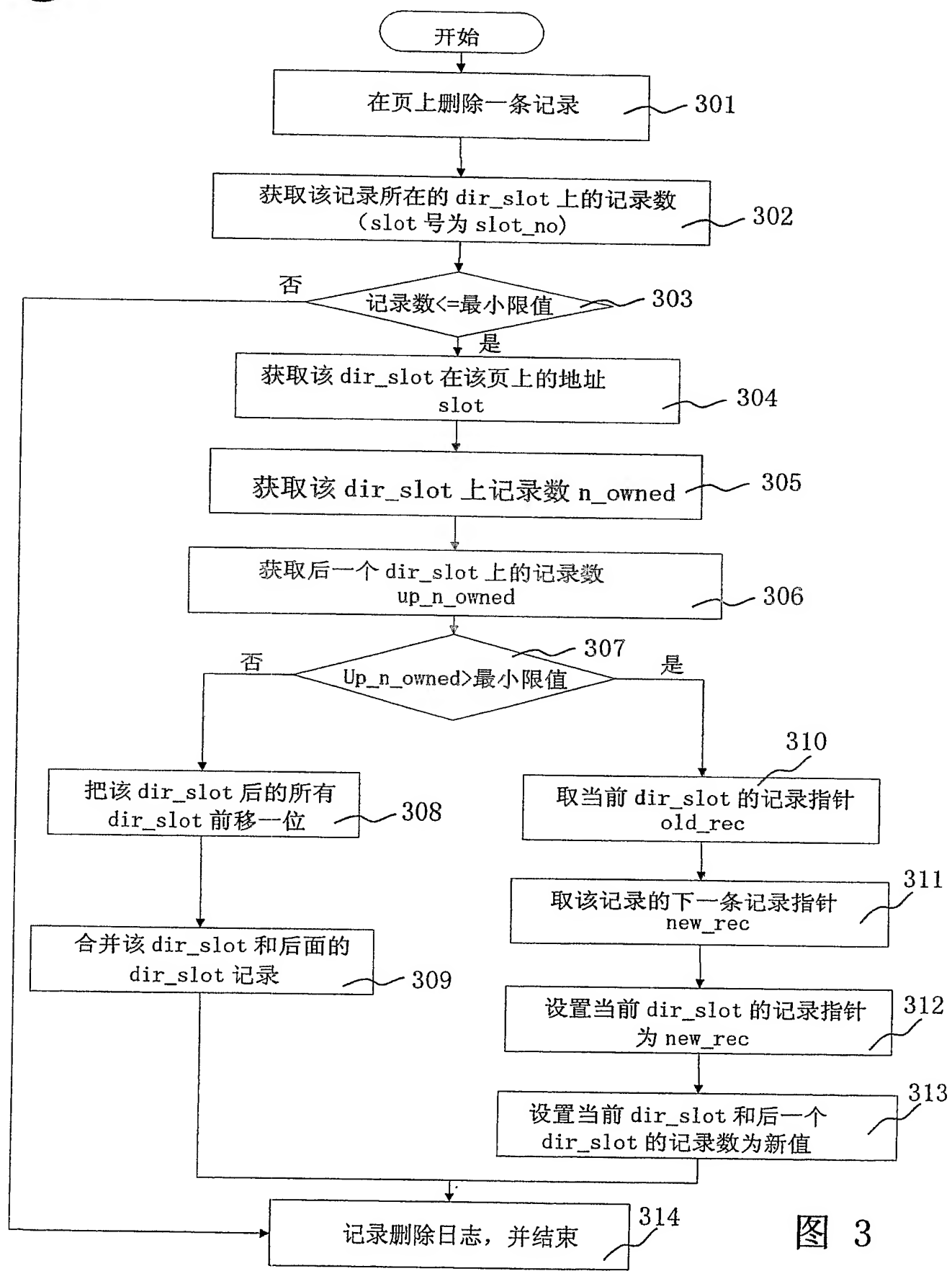


图 3

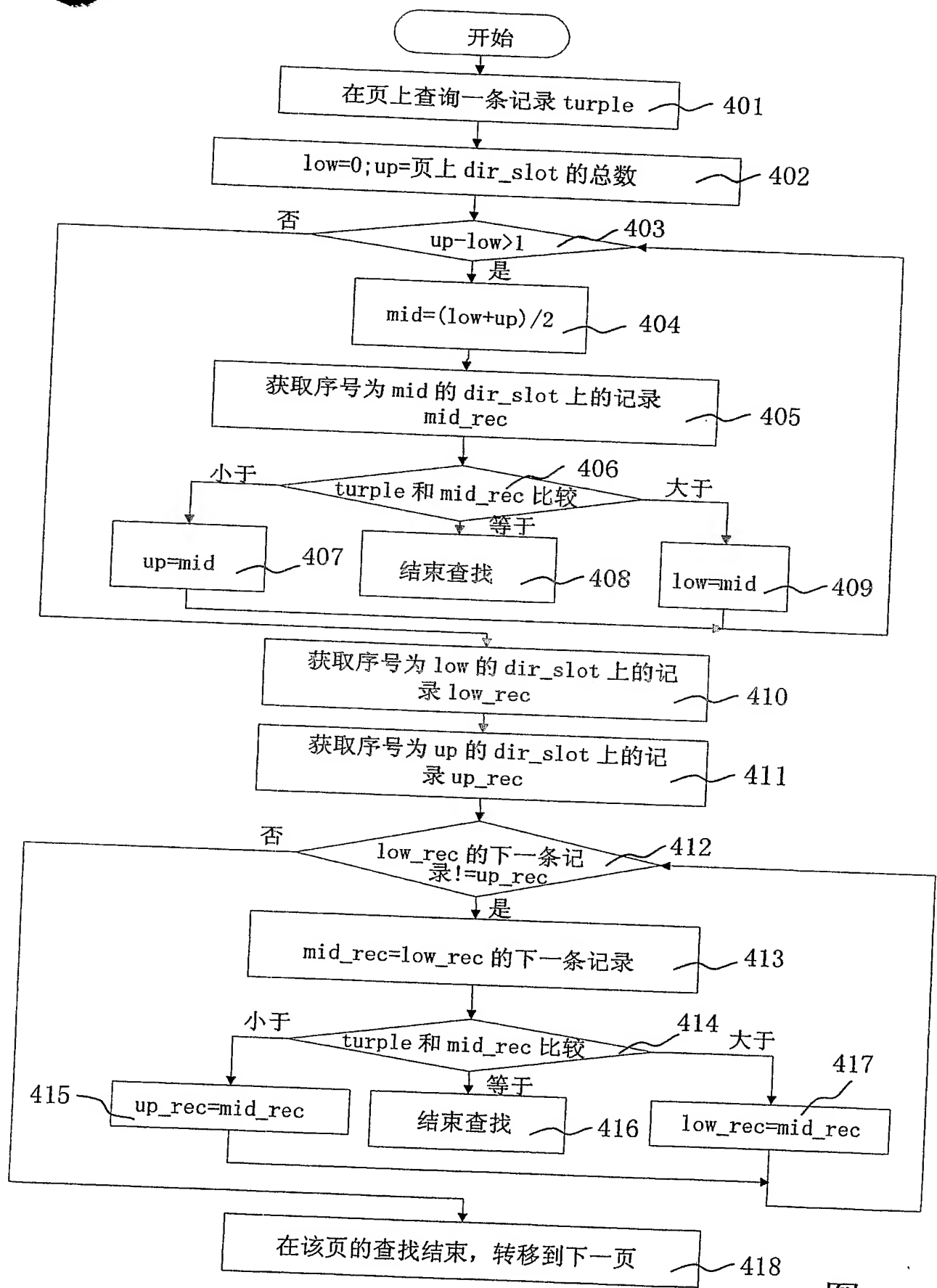


图 4